# CS103 Syllabus

This handout contains the tentative syllabus for CS103. Depending on how quickly we're able to cover various topics, we may proceed more quickly or more slowly than the syllabus indicates.

The readings from Sipser can be taken from either the second or third edition of the book.

| **Part One: Mathematical Formalisms** | | | |
|---|---|---|---|
| **Date** | **Topics** | **Readings** | **Assignments** |
| **M** Sep 24 | *Can computers solve all problems?*<br>Set Theory<br>The Limits of Computing | Course Notes Ch. 1 | |
| **W** Sep 26 | *How do we prove results with certainty?*<br>Direct Proofs | Course Notes Ch. 2 | |
| **F** Sep 28 | *How do we prove something without directly proving it?*<br>Proof by Contradiction<br>Proof by Contrapositive | Course Notes Ch. 2 | PS1 Out |
| **M** Oct 1 | *How do we reason about discrete structures?*<br>Mathematical Induction | Course Notes Ch. 3 | PS1 Checkpoint Due |
| **W** Oct 3 | *How else might we reason about discrete structures?*<br>Strong Induction<br>The Well-Ordering Principle | Course Notes Ch. 3 | |
| **F** Oct 5 | *How can we model connections between objects?*<br>Graphs<br>Relations | Course Notes Ch. 4, Ch. 5 | PS1 Due<br>PS2 Out |
| **M** Oct 8 | *How do we reason about the sizes of infinite sets?*<br>Functions<br>Cantor's Theorem Revisited | Handouts | PS2 Checkpoint Due |
| **W** Oct 10 | *Can we prove results simply by counting?*<br>The Pigeonhole Principle<br>Algorithmic Lower Bounds | Handouts | |
| **F** Oct 12 | *How do we symbolically codify logic?*<br>Propositional Logic | Rosen 1.1-1.2 | PS2 Due<br>PS3 Out |
| **M** Oct 15 | *What are the laws of mathematical reasoning?*<br>First-Order Logic | Rosen 1.3-1.5 | PS3 Checkpoint Due |
| **W** Oct 17 | *How do computers prove simple theorems?*<br>SAT Solving<br>Propositional Normal Forms | | |

| Part Two: Computability Theory | | | |
|---|---|---|---|
| **Date** | **Topics** | **Readings** | **Assignments** |
| **F** Oct 19 | *How do we mathematically model computers?*<br>Introduction to Computability Theory<br>DFAs | Sipser 1.1 | PS3 Due, PS4 Out |
| **M** Oct 22 | *Does computation have to be deterministic?*<br>NFAs<br>Equivalence of DFAs and NFAs | Sipser 1.2 | PS4 Checkpoint Due |
| **W** Oct 24 | *Can we assemble programs out of smaller programs?*<br>Regular Expressions<br>Equivalence of Regular Expressions and NFAs | Sipser 1.3 | |
| **F** Oct 26 | *Can computers with finite memory solve all problems?*<br>Non-Regular Languages<br>The Pumping Lemma for Regular Languages | Sipser 1.4 | PS4 Due, PS5 Out |
| **M** Oct 29 | *How are natural languages and formal languages alike?*<br>Context-Free Grammars<br>Context-Free Languages | Sipser 2.1 | |
| | **Midterm Exam: 7PM – 10PM**, Location TBA | | |
| **W** Oct 31 | *What might computers with unbounded memory look like?*<br>Pushdown Automata<br>Deterministic Context-Free Languages | Sipser 2.2 | |
| **F** Nov 2 | *How powerful is stack-based memory? What comes next?*<br>The Pumping Lemma for Context-Free Languages<br>Turing Machines | Sipser 2.3 | PS5 Due, PS6 Out |
| **M** Nov 5 | *How powerful is the Turing machine?*<br>A Programming Language for Turing Machines<br>The Church-Turing Thesis | Sipser 3.1 | |
| **W** Nov 7 | *Can machines perform computations on other machines?*<br>The Universal Turing Machine<br>Nondeterministic Turing Machines | Sipser 3.2–3.3 | |
| **F** Nov 9 | *Why are threads and recursion possible?*<br>A Theoretical Basis for Systems Programming | | PS6 Due, PS7 Out |
| **M** Nov 12 | *What problems cannot be solved?*<br>An Unrecognizable Language<br>An Undecidable Language | Sipser 4.2 | PS7 Checkpoint Due |
| **W** Nov 14 | *How do we learn about one problem by studying another?*<br>Mapping Reductions | Sipser 5.1 | |
| **F** Nov 16 | *What is the maximum extent of computing power?*<br>More Mapping Reductions<br>co-**RE** Languages | Sipser 5.3 | PS7 Due, PS8 Out |
| **Thanksgiving Break**<br>**November 19 – 23**<br>**No Class** | | | |

| Part Three: Complexity Theory | | | |
|---|---|---|---|
| **Date** | **Topics** | **Readings** | **Assignments** |
| **M** Nov 26 | *What problems can be solved efficiently?*<br>Introduction to Complexity Theory<br>**P** | Sipser 7.1–7.3 | PS8 Checkpoint Due |
| **W** Nov 28 | *What problems can be verified efficiently?*<br>**NP**<br>**P** $\stackrel{?}{=}$ **NP** | Sipser 7.4 | |
| **F** Nov 30 | *What are the hardest problems to verify efficiently?*<br>**NP**-Completeness<br>The Cook-Levin Theorem | Sipser 7.5 | PS8 Due, PS9 Out |
| **M** Dec 3 | *How can you solve one problem with a solver for another?*<br>More **NP**-Completeness | | |
| **W** Dec 5 | *Are all hard problems created equal?*<br>Approximation Algorithms<br>**NP**-Completeness and Cryptography | | |
| **F** Dec 7 | *How does everything fit together?*<br>The Big Picture | | PS9 Due |
| **W** Dec 12 | **Final Exam: 12:15 – 3:15PM**, Location TBA | | |